

Amendments to the Claims:

Applicant reserves the right to pursue any canceled claims at a later date. The following listing of claims will replace all prior versions, and listings, of claims in the application:

1-24. (canceled)

25. (currently amended) A method for executing a program for an industrial automation system, comprising:

providing a computer unit with:

input aids,

output aids.

a display device,

modules and functions respectively representing sub-tasks of an automation solution, and

a program which is structured from the modules and functions;

converting the modules and functions of the structured program into [[to]] objects to create a machine-independent program in the form of a hierarchical tree; and

loading the machine-independent program in the form of the at least one hierarchical tree into the corresponding components of the automation system,

wherein the corresponding components of the automation system execute the machine-independent program present in the form of the at least one hierarchical tree with the aid of at least one object machine assigned to the corresponding components of the automation system, and wherein the at least one object machine provides operators and objects from which the machine-independent program is provided in the form of the at least one hierarchical tree; and

during or after loading of the machine-independent program, instantiating the operators using the at least one object machine into corresponding components of the automation system; and

converting the symbolic representation of the hierarchical tree to physical addresses to generate a loadable program in the form of an executable program or operator tree.

26. (canceled)

27. (previously presented) The method according to claim 25, wherein the objects of the machine-independent program are present in the form of at least one hierarchical object or operator tree in the corresponding components of the automation system and are processed interpretatively.

28. (previously presented) The method according to claim 27, wherein the machine-independent program is present in the form of at least one object or operator tree with a structure equivalent or similar to the representation of the program in the display device.

29. (previously presented) The method according to claim 25, wherein the machine-independent program is loaded into the corresponding components of the automation system using a machine-independent, symbolic representation of the hierarchical tree.

30. (currently amended) The method according to claim 29, wherein the machine-independent and symbolic representation of the hierarchical tree is in the form of a byte code language or a markup language ~~such as extended markup language~~.

31. (previously presented) The method according to claim 25, wherein the object machine is configured as a real-time object machine with deterministic response and cycle times.

32-33. (canceled)

34. (previously presented) The method according to claim 25, wherein the object machine is implemented as a function unit that is closed and that processes the at least one hierarchical tree to a runtime system of the automated system.

35. (previously presented) The method according to claim 27, wherein the object machine is implemented in a distributed manner as at least one object, with the hierarchical object or operator tree processing itself.

36. (previously presented) The method according to claim 25, wherein the modules and functions are assigned model information and/or meta-information using the input aids and/or the display device.

37. (previously presented) The method according to claim 27, wherein the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services or infrastructure functions that access the objects via containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects.

38. (previously presented) A device for executing a program for an industrial automation system, comprising:

at least one computer unit with input aids, output aids and a display device;

a component for modeling and/or creating modules and functions, which respectively represent the sub-tasks of an automation solution;

a component for structuring the modules and functions and for networking the same, to form at least one hierarchical tree as at least one machine-independent program; and

a component to load the machine-independent program in the form of the at least one hierarchical tree into the corresponding components of the automation system with the corresponding components of the automation system executing the machine-independent program present in the form of the at least one hierarchical tree, wherein at least one object machine is assigned to the corresponding components of the automation system to execute the machine-independent programs, and wherein the at least one object machine provides operators and objects from which the machine-independent program is provided in the form of the hierarchical tree;

a component to instantiate the operators using the at least one object machine during or after the loading of the machine-independent program into corresponding components of the automation system; and

a component to convert the symbolic representation of the at least one hierarchical tree to physical addresses to generate a loadable program in the form of an executable program or operator tree.

39. (canceled)

40. (previously presented) The device according to claim 38, wherein the machine-independent program is present in the form of at least one object or operator tree with a structure equivalent or similar to the representation of the program in the display device.

41. (previously presented) The device according to claim 38, wherein the at least one object machine is configured as a real-time object machine with deterministic response and cycle times.

42. (canceled)

43. (previously presented) The device according to claim 38, further comprising a device for assigning model information and/or meta-information to the modules and functions.

44. (previously presented) The device according to claim 38, wherein the object machine is implemented as a function unit that is closed and processes the at least one hierarchical tree to a runtime system of the automated invention.

45. (previously presented) The device according to claim 38, wherein the object machine is implemented in a distributed manner as at least one object, with the hierarchical object or operator tree processing itself.

46. (previously presented) The device according to claim 38, wherein the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services or infrastructure functions that access the objects via containers assigned to the objects such that an infrastructure service or infrastructure function can be used by all the objects.

47. (previously presented) A computer program implementing a method for executing a program for an industrial automation system, comprising:

providing a computer unit with input aids, output aids and a display device, having modules and functions respectively representing sub-tasks of an automation solution being modeled and/or created using the input aids and optionally the display device, having the modules and functions being structured and networked using the input aids and optionally the display device so as to form a hierarchical tree as a machine-independent program;

loading the machine-independent program in the form of the hierarchical tree into the corresponding components of the automation system, wherein the corresponding components of the automation system execute the machine-independent program present in the form of the hierarchical tree with the aid of at least one object machine assigned to the corresponding components of the automation system, and wherein the at least one object machine provides operators and objects from which the machine-independent program is provided in the form of the hierarchical tree;

during or after loading of the machine-independent program, instantiating the operators using the at least one object machine into corresponding components of the automation system; and

converting the symbolic representation of the hierarchical tree to physical addresses to generate a loadable program in the form of an executable program or operator tree.